Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) An animation deformation pipeline embodied in a computer-readable storage medium, comprising:

a head node embodied in a computer-readable medium, the head node configured to convert a geometric representation of an object into a data stream;

a plurality of deformation nodes embodied in a computer-readable medium, each deformation node configured to receive the data stream from a node, and for applying a deformation to the data stream;

a tail node embodied in a computer-readable medium, the tail node configured to convert the deformed data stream into a geometric representation of a deformed object, wherein topology information describing the topology makeup of the geometry encoded in the data stream is passed through the plurality of deformation nodes separate from the data stream without altering the topology information.

2. (Original) The animation deformation pipeline of claim 1, wherein at least a subset of the deformation nodes apply deformations in response to manipulation of a polygonal proxy model.

3. (Original) The animation deformation pipeline of claim 1, wherein each deformation node passes the data stream to a succeeding node.

4. (Original) The animation deformation pipeline of claim 1, wherein at least a subset of the deformation nodes apply deformations using a sequential binding mode.

5. (Original) The animation deformation pipeline of claim 4, wherein at least a subset of the deformation nodes apply deformations to the result of a previous deformation node.

6. (Original) The animation deformation pipeline of claim 4, wherein each deformation node receives, within the data stream, a representation of a point, deforms the point, and passes, to a succeeding node, a representation of the deformed point.

7. (Original) The animation deformation pipeline of claim 1, wherein at least a subset of the deformation nodes apply deformations using a parallel binding mode.

8. (Original) The animation deformation pipeline of claim 7, wherein at least a subset of the deformation nodes apply deformations by combining influences of at least two polygonal proxy models.

9. (Original) The animation deformation pipeline of claim 7, wherein each deformation node receives, within the data stream, a representation of a point, deforms the point, and passes, to a succeeding node, a representation of the undeformed point.

10. (Original) The animation deformation pipeline of claim 7, wherein each deformation node receives, within the data stream, a representation of a point, deforms the point, and passes, to a succeeding node, a representation of the deformed point and a representation of the undeformed point.

11. (Original) The animation deformation pipeline of claim 1, wherein at least a subset of the deformation nodes apply deformations using a blend binding mode.

12. (Original) The animation deformation pipeline of claim 11, wherein at least a subset of the deformation nodes generate output that interpolates a current deformation with output of at least one other deformation node.

13. (Original) The animation deformation pipeline of claim 1, wherein at least a subset of the deformation nodes apply deformations using a hierarchical binding mode.

14. (Original) The animation deformation pipeline of claim 13, wherein at least a subset of the deformation nodes apply deformations to a local origin point of an input binding site.

15. (Original) The animation deformation pipeline of claim 1, wherein at least a subset of the deformation nodes are associated with a user-specifiable weight parameter that controls the relative amount of deformation applied by the node.

16. (Original) The animation deformation pipeline of claim 15, wherein each deformation node associated with a user-specifiable weight generates output representing a weighted combination of the input to the deformation node and the result of the deformation applied by the node.

17. (Original) The animation deformation pipeline of claim 15, wherein the weights are normalized over the entire pipeline.

18. (Original) The animation deformation pipeline of claim 15, wherein the weights are not normalized.

19. (Original) The animation deformation pipeline of claim 1, wherein each deformation node applies its deformation by:

determining a binding site for at least one control vertex of the object; transforming the binding site;

propagating the transformation of the binding site to the control vertex of the object, to establish a new location for the control vertex; and deforming the object according to the new location of the control vertex.

20. (Original) The animation deformation pipeline of claim 17, wherein the binding sites are locations in a subdivision surface.

21. (Original) The animation deformation pipeline of claim 19, wherein the binding sites are components of a polygonal proxy model, and wherein propagating the transformation comprises

deforming a subdivision surface, wherein the subdivision surface passes smooth deformations to the control vertices of the object.

22. (Original) The animation deformation pipeline of claim 1, wherein each deformation node generates data stream output for another node, the data stream output comprising a representation of the deformed object.

23. (Original) The animation deformation pipeline of claim 1, wherein the data stream comprises a plurality of binding items to be deformed by deformation nodes.

24. (Original) The animation deformation pipeline of claim 23, wherein the binding items comprise tags specifying binding modes.

25. (Original) The animation deformation pipeline of claim 24, wherein each deformation node has a binding mode, and wherein each deformation node applies deformations on binding items having a tag specifying a matching binding mode.

26. (Original) The animation deformation pipeline of claim 24, further comprising at least one filter node for modifying tags.

27. (Original) The animation deformation pipeline of claim 24, wherein at least one binding item comprises a tag specifying that no deformations are to be applied, and wherein the deformation nodes allow the binding item having the tag to pass without deformation.

28. (Original) The animation deformation pipeline of claim 24, further comprising at least one masking node for modifying tags.

29. (Original) The animation deformation pipeline of claim 28, wherein the masking node modifies a tag to specify that a binding item be excluded from deformation by a particular deformation node.

30. (Original) The animation deformation pipeline of claim 1, wherein each node comprises a graphics hardware component.

31. (Original) The animation deformation pipeline of claim 1, wherein: the object comprises a plurality of surfaces;

the data stream comprises at least one data block for each surface of the object;

and each deformation node applies a deformation by modifying at least one data block associated with the object surface being deformed.

32. (Original) The animation deformation pipeline of claim 31, wherein each surface is associated with a plurality of control vertices, and wherein each data block

comprises a binding item entry for each control vertex of the surface associated with the data block.

33. (Original) The animation deformation pipeline of claim 1, wherein each deformation node comprises a user-specifiable attribute for enabling and disabling the node.

34. (Original) The animation deformation pipeline of claim 1, wherein each deformation node comprises a user-specifiable attribute indicating a blending mode.

35. (Original) The animation deformation pipeline of claim 1, wherein each deformation node comprises a user-specifiable attribute indicating a weighting factor.

36. (Currently Amended) A method for deforming a computer-generated object using a deformation pipeline, comprising:

converting a geometric representation of an object into a data stream;

for each of a plurality of deformation nodes, receiving the data stream and applying a deformation to the data stream;

passing topology information describing the topology makeup of the geometry encoded in the data stream through the plurality of deformation nodes separate from the data stream without altering the topology information; ~~and~~

converting the deformed data stream into a geometric representation of a deformed object; and

displaying the geometric representation of the deformed object.

37. (Original) The method of claim 36, wherein for at least a subset of the deformation nodes, applying a deformation comprises applying the deformation in response to manipulation of a polygonal proxy model.

38. (Original) The method of claim 36, further comprising, for each of the deformation nodes, passing the data stream to a succeeding node.

39. (Original) The method of claim 36, wherein for at least a subset of the deformation nodes, applying a deformation comprises applying the deformation using a sequential binding mode.

40. (Original) The method of claim 39, wherein for at least a subset of the deformation nodes, applying a deformation comprises applying the deformation to the result of a previous deformation node.

41. (Original) The method of claim 39, further comprising, for each of the deformation nodes, receiving within the data stream a representation of a point, deforming the point, and passing, to a succeeding node, a representation of the deformed point.

42. (Original) The method of claim 36, wherein for at least a subset of the deformation nodes, applying deformations comprises using a parallel binding mode.

43. (Original) The method of claim 42, wherein for at least a subset of the deformation nodes, applying a deformation comprises combining influences of at least two polygonal proxy models.

44. (Original) The method of claim 42, wherein each deformation node receives, within the data stream, a representation of a point, deforms the point, and passes, to a succeeding node, a representation of the undeformed point.

45. (Original) The method of claim 42, further comprising, for each of the deformation nodes, receiving, within the data stream, a representation of a point, deforming the point, and passing, to a succeeding node, a representation of the deformed point and a representation of the undeformed point.

46. (Original) The method of claim 36, wherein for at least a subset of the deformation nodes, applying a deformation comprises applying deformations using a blend binding mode.

47. (Original) The method of claim 46, further comprising, for at least a subset of the deformation nodes interpolating a current deformation with output of at least one other deformation node.

48. (Original) The method of claim 36, wherein for at least a subset of the deformation nodes, applying a deformation comprises applying deformations using a hierarchical binding mode.

49. (Original) The method of claim 48, wherein for at least a subset of the deformation nodes, applying a deformation comprises applying deformations to a local origin point of an input binding site.

50. (Original) The method of claim 36, further comprising, for at least a subset of the deformation nodes, receiving a user-specifiable weight parameter controlling the relative amount of deformation applied by the node.

51. (Original) The method of claim 50, further comprising, for at least a subset of the deformation nodes, generating output representing a weighted combination of the input to the deformation node and the result of the deformation applied by the node.

52. (Original) The method of claim 50, further comprising normalizing the weights over the entire pipeline.

53. (Original) The method of claim 36, wherein, for at least a subset of the deformation nodes, applying a deformation node comprises:

determining a binding site for at least one control vertex of the object;

transforming the binding site;

propagating the transformation of the binding site to the control vertex of the object, to establish a new location for the control vertex; and deforming the object according to the new location of the control vertex.

54. (Original) The method of claim 53, wherein the binding sites are locations in a subdivision surface.

55. (Original) The method of claim 53, wherein the binding sites are components of a polygonal proxy model, and wherein propagating the transformation comprises deforming a subdivision surface, wherein the subdivision surface passes smooth deformations to the control vertices of the object.

56. (Original) The method of claim 36, further comprising, for each deformation node, generating data stream output for another node, the data stream output comprising a representation of the deformed object.

57. (Original) The method of claim 36, wherein the data stream comprises a plurality of binding items to be deformed by deformation nodes.

58. (Original) The method of claim 57, wherein the binding items comprise tags specifying binding modes.

59. (Original) The method of claim 58, wherein each deformation node has a binding mode, and wherein, for each deformation node, applying a deformation comprises applying a deformation on binding items having a tag specifying a matching binding mode.

60. (Original) The method of claim 36, wherein:

the object comprises a plurality of surfaces;

the data stream comprises at least one data block for each surface of the object;

and wherein, for each deformation node, applying a deformation comprises modifying at least one data block associated with the object surface being deformed.

61. (Original) The method of claim 60, wherein each surface is associated with a plurality of control vertices, and wherein each data block comprises a binding item entry for each control vertex of the surface associated with the data block.

62. (Original) The method of claim 36, further comprising, for each deformation node, receiving a user-specifiable attribute for enabling and disabling the node.

63. (Original) The method of claim 36, further comprising, for each deformation node, receiving a user-specifiable attribute indicating a blending mode.

64. (Original) The method of claim 36, further comprising, for each deformation node, receiving a user-specifiable attribute indicating a weighting factor.

65. (Currently Amended) A computer program product for deforming a computer-generated object using a deformation pipeline, comprising:

a computer-readable medium; and

computer program code, encoded on the medium, for:

converting a geometric representation of an object into a data stream;

for each of a plurality of deformation nodes, receiving the data stream and applying a deformation to the data stream;

passing topology information describing the topology makeup of the geometry encoded in the data stream through the plurality of deformation nodes separate from the data stream without altering the topology information; and

converting the deformed data stream into a geometric representation of a deformed object.

66. (Original) The computer program product of claim 65, wherein for at least a subset of the deformation nodes, the computer program code for applying a deformation comprises computer program code for applying the deformation in response to manipulation of a polygonal proxy model.

67. (Original) The computer program product of claim 65, further comprising computer program code for, for each of the deformation nodes, passing the data stream to a succeeding node.

68. (Original) The computer program product of claim 65, wherein for at least a subset of the deformation nodes, the computer program code for applying a deformation comprises computer program code for applying the deformation using a sequential binding mode.

69. (Original) The computer program product of claim 68, wherein for at least a subset of the deformation nodes, the computer program code for applying a deformation comprises computer program code for applying the deformation to the result of a previous deformation node.

70. (Original) The computer program product of claim 68, further comprising, for each of the deformation nodes, computer program code for receiving within the data stream a

representation of a point, deforming the point, and passing, to a succeeding node, a representation of the deformed point.

71. (Original) The computer program product of claim 65, wherein for at least a subset of the deformation nodes, the computer program code for applying deformations uses a parallel binding mode.

72. (Original) The computer program product of claim 71, wherein for at least a subset of the deformation nodes, the computer program code for applying a deformation comprises computer program code for combining influences of at least two polygonal proxy models.

73. (Original) The computer program product of claim 71, wherein each deformation node receives, within the data stream, a representation of a point, deforms the point, and passes, to a succeeding node, a representation of the undeformed point.

74. (Original) The computer program product of claim 71, further comprising, for each of the deformation nodes, computer program code for receiving, within the data stream, a representation of a point, deforming the point, and passing, to a succeeding node, a representation of the deformed point and a representation of the undeformed point.

75. (Original) The computer program product of claim 65, wherein for at least a subset of the deformation nodes, the computer program code for applying a deformation comprises computer program code for applying deformations using a blend binding mode.

76. (Original) The computer program product of claim 75, further comprising computer program code for, for at least a subset of the deformation nodes interpolating a current deformation with output of at least one other deformation node.

77. (Original) The computer program product of claim 65, wherein for at least a subset of the deformation nodes, the computer program code for applying a deformation comprises computer program code for applying deformations using a hierarchical binding mode.

78. (Original) The computer program product of claim 77, wherein for at least a subset of the deformation nodes, the computer program code for applying a deformation comprises computer program code for applying deformations to a local origin point of an input binding site.

79. (Original) The computer program product of claim 65, further comprising, for at least a subset of the deformation nodes, computer program code for receiving a user-specifiable weight parameter controlling the relative amount of deformation applied by the node.

80. (Original) The computer program product of claim 79, further comprising, for at least a subset of the deformation nodes, computer program code for generating output representing a weighted combination of the input to the deformation node and the result of the deformation applied by the node.

81. (Original) The computer program product of claim 79, further comprising computer program code for normalizing the weights over the entire pipeline.

82. (Original) The computer program product of claim 65, wherein, for at least a subset of the deformation nodes, the computer program code for applying a deformation node comprises computer program code for:

determining a binding site for at least one control vertex of the object; transforming the binding site;

propagating the transformation of the binding site to the control vertex of the object, to establish a new location for the control vertex; and deforming the object according to the new location of the control vertex.

83. (Original) The computer program product of claim 82, wherein the binding sites are locations in a subdivision surface.

84. (Original) The computer program product of claim 82, wherein the binding sites are components of a polygonal proxy model, and wherein the computer program code for propagating

the transformation comprises computer program code for deforming a subdivision surface, wherein the subdivision surface passes smooth deformations to the control vertices of the object.

85. (Original) The computer program product of claim 65, further comprising, for each deformation node, computer program code for generating data stream output for another node, the data stream output comprising a representation of the deformed object.

86. (Original) The computer program product of claim 65, wherein the data stream comprises a plurality of binding items to be deformed by deformation nodes.

87. The computer program product of claim 86, wherein the binding items comprise tags specifying binding modes.

88. (Original) The computer program product of claim 87, wherein each deformation node has a binding mode, and wherein, for each deformation node, the computer program code for applying a deformation comprises computer pro gram code for applying a deformation on binding items having a tag specifying a matching binding mode.

89. (Original) The computer program product of claim 65, wherein: the object comprises a plurality of surfaces;

the data stream comprises at least one data block for each surface of the object;

and wherein, for each deformation node, the computer program code for applying a deformation comprises computer program code for modifying at least one data block associated with the object surface being deformed.

90. (Original) The computer program product of claim 89, wherein each surface is associated with a plurality of control vertices, and wherein each data block comprises a binding item entry for each control vertex of the surface associated with the data block.

91. (Original) The computer program product of claim 65, further comprising, for each deformation node, computer program code for receiving a user-specifiable attribute for enabling and disabling the node.

92. (Original) The computer program product of claim 65, further comprising, for each deformation node, computer program code for receiving a user-specifiable attribute indicating a blending mode.

93. (Original) The computer program product of claim 65, further comprising, for each deformation node, computer program code for receiving a user-specifiable attribute indicating a weighting factor.